Leveraging 1-hop Neighborhood Knowledge for Efficient Flooding in Mobile Ad Hoc Networks

Ying Cai * *Kien A. Hua* † *Guohong Cao* ‡

February 8, 2004

Abstract

Mobile Ad hoc Network (MANET) has attracted great research interest due to its simplicity and low cost of deployment. The ever-changing network topology, however, presents many new challenges to the existing communication protocols well developed for networks with some fixed infrastructure. Without the presence of any fixed central servers, flooding is a fundamental and frequently invoked communication primitive in mobile ad hoc networks. For example, it is often used to discover network topology information (e.g., host locations, etc.) for routing purpose. Unfortunately, existing flooding techniques either generate enormous amount of redundant broadcast retransmission or incur excessive network control overhead. To address these crucial problems, we propose a technique called *Edge Forwarding*. This new scheme minimizes the flooding traffic by leveraging location information to limit broadcast retransmissions to only hosts near the perimeter of each broadcast coverage. Unlike most existing techniques, Edge forwarding requires each host to track only neighboring nodes within its one-hop distance. Therefore, it can be easily incorporated into many existing routing protocols without incurring any additional control overhead. Our performance studies indicate that with our strategy, a substantial portion of the unnecessary broadcast retransmissions can be eliminated.

KEYWORDS: ad hoc network, flooding, mobile computing, radio transmission range.

*Department of Computer Science, Iowa State University, Ames, IA 50011, E-mail: yingcai@cs.iastate.edu

†School of Computer Science, University of Central Florida, Orlando, FL 32816, E-mail: kienhua@cs.ucf.edu

‡Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802, E-mail: gcao@cse.psu.edu

1 Introduction

A *Mobile Ad hoc Network* (MANET) consists of a number of mobile hosts that connect each other through wireless communication. In MANET, the end-to-end communication relies solely on the radio packet relaying among the participating hosts. As the radio transmission range of a wireless host is very limited, a packet typically has to go through many hops before reaching its destination. Therefore, every mobile host in such a network has the obligation to behave on demand as a router to ensure packet delivery.

MANET has received great research interest due to its simplicity and low cost of deployment. Unlike the existing cellular networks, the communication in MANET does not depend on any fixed networking infrastructure. Therefore, it can be built conveniently. In the places where a wired backbone network is neither available nor economical to build, MANET could play a critical role. Some application scenarios include law enforcement operations, battlefield communications, disaster recovery, entertainment, and so on.

In MANET, the network topology changes rapidly as a result of host mobility. In such an environment, flooding, i.e., broadcasting a message to all hosts, is a fundamental and frequently invoked communication primitive. For example, it is essential to ad hoc routing algorithms (e.g. DSR [1], AODV [2], ZRP [3], LAR [4], etc.) for route discovery. A simple flooding technique is as follows:

1. the sender broadcasts the packet to its 1-hop neighbors [1];
2. upon receiving a flooding packet, a host rebroadcasts it if the packet has not been received before.

The above *plain flooding* [5] [6] guarantees that a data packet can reach all hosts that

[1]Two hosts are said to be n-hop neighbors if they can communicate with each other over n wireless links

are reachable from its originator if no packet collision occurs. This simple strategy, however, generates overwhelming network traffic: all hosts have to transmit the data packet once; and each host typically receives the same broadcast many times. In the worst case, a flooding packet can travel each wireless link twice. This strategy is too inefficient with respect to energy used as data communications are known to consume the most battery power of the mobile hosts compared to other computing operations [7] [8]. In addition, the entire network could also hang up as a result of severe packet contention and collision [9]. Furthermore, if the hosts have to compete for limited communication bandwidth, then the excessive network traffic could cause a significant delay in packet transmission.

To reduce the flooding traffic, many approaches have been proposed. We classify these techniques into three categories and discuss their limitation as follows:

- *0-hop schemes*: Many flooding techniques developed in early time are in this category. These techniques try to reduce flooding cost without any assumption on neighborhood knowledge. For example, one can simply make each host rebroadcast flooding packets with some predetermined probability. This probabilistic-based scheme was first proposed in [9] [10] and further investigated in [11]. Other approaches proposed in [9], including counter-based, distance-based, location-based, and cluster-based flooding schemes, are also in this category. These techniques can effectively reduce the flooding traffic and has the same implementation cost as plain flooding. Their efficiency, however, is achieved at the expense of flooding reachability. In these schemes, a non-redundant retransmission might be dropped in the first few hops; and its effect could propagate to the following hops causing the number of unreachable hosts to amplify quickly hop after hop. Besides the reachability issue, another major problem of these techniques is that it is very difficult to find a threshold value (e.g., retransmission probability, etc.) appropriate for various network situations [12].

- *1-hop schemes*: The technique called Flooding with Self Pruning (FSP) in [13] is a 1-hop flooding scheme since it requires each host to track its neighbors within 1-hop distance. In this scheme, when a host broadcasts a packet, it includes all of its 1-hop neighbors in the packet header. Upon receiving a broadcast, a host checks its own 1-hop neighbors and if all of them have already been listed in the broadcast packet header, it does not forward the broadcast. Although this technique can reduce the flooding cost and guarantee packet reachability at the same time, its performance improvement is very limited in most network conditions [14].

- *2^+-hop schemes*: Most existing flooding approaches are in this category and they can be further divided into *reactive* schemes and *proactive* schemes. In proactive schemes [13] [15] [16] [17], a broadcasting host selects some of its 1-hop neighbors as rebroadcasting hosts. When a host receives a broadcast, it drops off the packet if it is not designated as a rebroadcasting host; otherwise, it recursively chooses some of its 1-hop neighbors as rebroadcasting hosts and then forwards the broadcast. In reactive schemes [18] [19] [20] [21] [22] [23] [24] [25], each host determines by its own on whether or not to forward a broadcast packet. In general, these techniques are not adaptive to networks with high mobility and host density. This is due to the fact that when the network topology changes frequently, the overhead of discovering and maintaining local network topology (within two or more hops) for each host increases, and may outweigh the benefit of reduction in retransmission [11] [12]. Furthermore, for those proactive techniques, the task of selecting a suitable set of hosts to forward the broadcasts is not trivial and requires significant computation on the mobile hosts. It was proven in [17] that finding the optimal set of rebroadcasting hosts is NP-hard.

In summary, recent techniques, as discussed above, either compromise flooding reachability, are inefficient in reducing network traffic, or require significant control overhead and intensive computation on mobile host.

In this paper, we address the aforementioned problems by considering a novel technique called *Edge Forwarding*. In this approach, we divide the transmission coverage of each host into six equal partitions. When a host receives a broadcast, it decides on its own whether or not to forward the broadcast using two different forwarding rules. Under the first rule, a host does not forward a broadcast unless it is *close* to some partition edge of the broadcast coverage. Thus, a significant percentage of redundant broadcast retransmission is eliminated. The second rule further reduces the unnecessary rebroadcast by pushing the forwarding responsibility to the hosts close to the perimeter of the broadcast coverage. While this new technique guarantees the desired flooding reachability, it is highly efficient and scalable to the network size and density. This is due to the fact that the retransmission of a broadcast typically occurs only at the perimeter of the broadcast coverage. Another major advantage of Edge Forwarding is its simplicity: it requires each host to know only its 1-hop neighbors, a prerequisite information to many existing routing protocols (e.g., [4], [30], [31], [32], etc.). Therefore, unlike the numerous 2^+-hop flooding techniques mentioned above, Edge Forwarding can be easily incorporated into these routing protocols to boost their performance without incurring any additional network control overhead.

We note that significant research has also been done recently on minimizing energy consumption of broadcasting over *static* wireless networks [26] [27] [28] [29]. These works focus on how to select the rebroadcasting hosts and adjust their transmission power (i.e., radius) so that the total energy consumed for flooding a data packet can be minimized. Since these schemes do not consider networks with mobility, and require each host to have complete knowledge of the fixed network topology, it is unlikely that they can be used in mobile ad hoc networks.

The remainder of this paper is organized as follows. We present Edge Forwarding concept in Section 2, and introduce the protocol in Section 3. In Section 4, we describe the simulation model and examine the performance results. Finally, we give our concluding remarks in Section 5.

2 Preliminary

2.1 Transmission Coverage Partitioning

Given a host, say A, we partition its transmission coverage into six equal-size regions and identify them as A_{P_1}, A_{P_2}, A_{P_3}, A_{P_4}, A_{P_5}, and A_{P_6}, respectively. The partitioning and naming rules are illustrated in Figure 1(a). We say a host is A's P_i neighbor, if the host is currently inside partition A_{P_i}, where $1 \leq i \leq 6$. For instance, in Figure 1(b), B is A's P_1 neighbor as B is located in A_{P_1}. Given A at location (x_a, y_a) and its 1-hop neighbor B at location (x_b, y_b), the distance between A and B is $dist(A, B) = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}$. We can determine which partition of A contains B with some simple computation as follows:

- if $x_a \leq x_b$ and $y_a \leq y_b$, then B is in A_{P_1} if $\frac{x_b - x_a}{dist(A,B)} >= \frac{1}{2}$; otherwise, B is in A_{P_2};
- if $x_a > x_b$ and $y_a < y_b$, then B is in A_{P_2} if $\frac{x_a - x_b}{dist(A,B)} \leq \frac{1}{2}$; otherwise, B is in A_{P_3};
- if $x_a > x_b$ and $y_a > y_b$, then B is in A_{P_4} if $\frac{x_a - x_b}{dist(A,B)} >= \frac{1}{2}$; otherwise, B is in A_{P_5};
- if $x_a \leq x_b$ and $y_a >= y_b$, then B is in A_{P_5} if $\frac{x_b - x_a}{dist(A,B)} \leq \frac{1}{2}$; otherwise, B is in A_{P_6};

In the next, we present two *forwarding rules*, by which a host can determine whether or not it should forward a broadcast packet. When a host forwards a broadcast, it adds its ID to the packet header to inform each recipient of the sender of this broadcast. In our discussion, we assume each host knows the accurate position of its 1-hop neighboring hosts. In reality, the location information typically includes some amount of error, as a result of host movement and the inaccuracy caused by the underlying positioning systems. The ideas suggested here, however, can be applied in general - the issue of location uncertainty will be addressed in the next section, where we present our new flooding protocol.

2.2 Basic Forwarding

Under this rule, when a host, say B, receives a new broadcast from another host, say A, B first determines which partition of A B is currently in. Given B in A_{P_i}, where $1 \leq i \leq 6$, it forwards the broadcast if there exists at least one partition B_{P_j}, where $1 \leq j \leq 6$, such that no other hosts can be found in $B_{P_j} \cap A_{P_i}$ (i.e., the overlapping area of B_{P_j} and A_{P_i}).

We use Figure 2 to explain the above forwarding rule. Without loss of generality, we assume B is in A_{P_1}. B's partition lines divide A_{P_1} into 6 subpartitions: $A_{P_{11}}$, $A_{P_{12}}$, $A_{P_{13}}$, $A_{P_{14}}$, $A_{P_{15}}$, and $A_{P_{16}}$, as showed in Figure 2. That is, $A_{P_{1i}} = A_{P_1} \cap B_{P_i}$, where $1 \leq i \leq 6$. When B receives a broadcast from A, B first determines if there is anyone of its P_1 neighbors inside $A_{P_{11}}$. This can be done by simply checking their distance to A: a B's P_1 neighbor, say H, is inside $A_{P_{11}}$ if and only if $dist(H, A) \leq R$, where R is A's transmission radius. If no host can be found in $A_{P_{11}}$, B forwards the broadcast. Otherwise, B continues to check its 1-hop neighbors in P_2, P_3, P_4, P_5, and P_6 sequentially. If there is at least one host in each of $A_{P_{1i}}$, where $1 \leq i \leq 6$, B does not forward the broadcast. The complexity of this procedure is $O(n)$, where n is the number of B's 1-hop neighbors. In practice, B can stop checking a subpartition $A_{P_{1i}}$ as soon as a host is found in this subpartition. Furthermore, if B detects an empty subpartition, B does not need to explore the remaining subpartitions. This strategy is, therefore, very energy efficient.

The above forwarding rule ensures that the broadcast can reach all of B's 1-hop neighbors that are outside of A's transmission coverage even if B does not forward the packet. In other words, at least one host in $A_{P_{1i}}$ will broadcast to cover B's P_i partition, where $1 \leq i \leq 6$. This can be proof as follows. Let's say B does not forward the broadcast in the above example. In this case, it means that there is at least one host in each of $A_{P_{1i}}$, where $1 \leq i \leq 6$. These hosts must have received the broadcast from A also since they are within 1-hop distance to A. Obviously, if there is a host inside $A_{P_{1i}}$ and it forwards the broadcast, then all hosts inside B_{P_i} will receive the broadcast. This is due to the fact that all B's P_i neighbors are within 1-hop distance to any host inside $A_{P_{1i}}$, as ensured by the nature of our partitioning. We now just need to prove that there is at least one host in each of $A_{P_{1i}}$, where $1 \le i \le 6$, will forward the broadcast.

We first prove that there is at least one host in $A_{P_{11}}$ that will forward the broadcast. Let's say host H^1 is currently inside $A_{P_{11}}$. Then similar to B, it divides A_{P_1} into 6 partitions. As H^1 is inside $A_{P_{11}}$, partition $H^1_{P_1} \cap A_{P_1}$ must be contained by $A_{P_{11}}$. If there is no host inside $H^1_{P_1} \cap A_{P_1}$, then according to the rule, H^1 must forward the broadcast. Otherwise, there must have at least one host inside $H^1_{P_1} \cap A_{P_1}$. Let host H^2 be such a host. Again, it divides A_{P_1} into 6 partitions and partition $H^2_{P_1} \cap A_{P_1}$ must be contained by $H^1_{P_1} \cap A_{P_1}$, because H^2 is inside $H^1_{P_1} \cap A_{P_1}$. If H^2 does not forward the broadcast, then let H^3 be a host inside $H^2_{P_1} \cap A_{P_1}$, ..., and so forth. These steps proceed recursively and eventually a host, say H^i, will find out that there is no hosts inside $H^i_{P_1} \cap A_{P_1}$ and need to forward the broadcast: as the value of i increases, the overlapping area $H^i_{P_1} \cap A_{P_1}$ becomes smaller and smaller and each time the number of hosts it contains, which is limited, is reduced at least by 1. Similarly, we can prove that in each of the remaining regions, $A_{P_{12}}$, $A_{P_{13}}$, $A_{P_{14}}$, $A_{P_{15}}$, and $A_{P_{16}}$, there is at least one host that will forward the broadcast.

2.3 Advanced Forwarding

As we can observe in Figure 2, a large portion of B's P_3 partition has already been covered by A's broadcast; and if there is any host inside the uncovered area, then very likely it is within 1-hop distance to all hosts inside $A_{P_{12}}$. If this is true, then it is not necessary to have some host inside $A_{P_{13}}$ to cover B_{P_3} partition. Obviously, partition B_{P_5} is in a similar situation. In addition, we do not need to consider the hosts inside B_{P_4} since this partition is completely covered by A's broadcast. Based on this observation, we develop an *advanced* forwarding rule - B does not need to forward a broadcast from A if the following three conditions are satisfied:

1. there is at least one host in each of $A_{P_{11}}$, $A_{P_{12}}$, and $A_{P_{16}}$;

2. all B's P_3 neighbors beyond 1-hop distance to A are within 1-hop distance to all hosts inside $A_{P_{12}}$;

3. all B's P_5 neighbors beyond 1-hop distance to A are within 1-hop distance to all hosts inside $A_{P_{16}}$.

We have already discussed how to determine if there is any host inside each of $A_{P_{1i}}$, where $1 \leq i \leq 6$. If there is no hosts in $A_{P_{13}}$, then we find out all B's P_3 neighbors whose distances to A are larger than the transmission radius R. For each of these hosts, say H_x, we check its distance to all hosts inside $A_{P_{12}}$. If there exists any host, say H_y, inside $A_{P_{12}}$, such that $dist(H_x, H_y) > R$, then the second condition is violated and B needs to forward the broadcast. The complexity of this procedure is $O(m * n)$, where m is the number of B's P_3 neighbors not covered by A's broadcast and n is the number of hosts inside $A_{P_{12}}$. If the second condition is satisfied, we then continue to check the third condition in a similar way.

We note that under the basic forwarding rule, a host does not need to forward a broadcast unless it is *close* to the edge of some broadcast partition. This characteristic eliminates a significant portion of unnecessary broadcast forwarding. The advanced forwarding rule enhances this by pushing the forwarding responsibility to only the hosts close to the transmission perimeter. This strategy, however, incurs more computation.

2.4 Handling Host Heterogeneity

The above discussion implicitly assumes that all hosts have the same transmission radius. In the presence of host heterogeneity, we can add a host's true transmission radius in its heartbeat broadcast and modify the basic forwarding rule as follows: Given host B in host A's partition

P_i, where $1 \leq i \leq 6$, B must forward a broadcast from A if there exists at least one B's partition B_{P_j}, where $1 \leq j \leq 6$, such that any one of the following conditions is satisfied:

- no hosts can be found in $B_{P_j} \cap A_{P_i}$;
- there is at least one host inside $B_{P_j} \cap A_{P_i}$ whose transmission cannot cover some of B's P_j neighbors beyond 1-hop distance to A.

Similarly, we can revise the advanced forwarding rule to handle heterogeneous hosts. Thus, our protocol presented in the next section can be used in general.

3 Proposed Technique: Edge Forwarding

In this section, we present our new flooding protocol and discuss its advantages.

3.1 Protocol Description

In Edge Forwarding, each flooding packet is associated with a life process. Upon receiving a flooding packet, a host spawns a new process to handle the packet if this packet has not been received before. The process first determines if it should forward the packet according to one of the above two forwarding rules and then puts itself into either one of the following two waiting scenarios. The waiting is an *overhearing* period if the underlying forwarding rule requires the host to rebroadcast the packet; otherwise, it is a *confirming* period. We will discuss shortly the settings of these two stages and the rationale behind them. During the waiting period, the host might receive the duplicated packets forwarded by other hosts. These packets are put into an internal queue of this process. At the end of the waiting period, the process checks the list of its 1-hop neighbors and for each one of them, determines if the neighbor can be reached by some broadcast in the queue. This can be done by simply calculating the distance between the neighbor and the sender of each broadcast. If there is at least one neighbor not covered by any received broadcasts, the process forwards the packet immediately and then terminates. This process is illustrated in a flowchart shown in Figure 3.

3.1.1 Setting Overhearing Period

We use the overhearing period to make broadcast retransmission occur in a more orderly manner. Given a broadcast, we prefer the hosts closer to its coverage perimeter to forward the broadcast in order to minimize the flooding. For example, in Figure 4, if both B and C are required by the forwarding rule to forward a broadcast from A, then C should forward the broadcast prior to B. Since B becomes aware of this forwarding, it might not need to forward the same packet.

In Edge Forwarding, each host sets and dynamically adjusts its overhearing period for a flooding packet. When a host A receives a new packet from another host B, A initializes the overhearing period for this packet to be $B.R - dist(A,B)$ time units [2]. Before the overhearing period expires, the host could receive duplicated packets from other hosts. For each of these packets, the overhearing period is adjusted as follows. If the duplicated packet arrives t time units later from a host C, where $0 \leq t \leq B.R - dist(A,B)$, host A adjusts its overhearing period for this packet to $max(B.R - dist(A,B) - t,\ C.R - dist(A,C) - t)$ time units.

We note that dynamic adjustment of overhearing period allows a host, say A, to further delay a packet forwarding in order to collect duplicated packets from more neighboring hosts. This increases the chance that all of A's neighbors have been covered by some previous broadcast of the same packet; and therefore A need not forward the packet. For example, in Figure 4, after host A broadcasts a packet, the initial forwarding order is host C, B, and then D. After

[2] Given a host, say H, its transmision radius is denoted as $H.R$

C forwards, the adjustment of the overhearing periods for the packet at host B and D makes D forward before B, since B is closer to C. Since B's transmission coverage is enclosed by the combined coverage of A, C, and D, B will not need to forward the packet.

3.1.2 Setting Confirming Period

In Edge Forwarding, we use confirming period to backup a broadcast retransmission in the case that an expected forwarding does not occur as a result of host uncertainty. Due to the continuous movement of mobile devices, the error caused by the underlying positioning systems, and other factors, it is possible that a host might have inaccurate position information about its neighbors. For instance, a host can be moving from one partition to another, and is momentarily not reachable for one heartbeat period. Other uncertainty, including host fragility and broadcast energy fading, could also make a host unreliable in terms of sending and receiving packet as expected.

If a host, say B, determines that it needs not forward a new flooding packet from another host, say A, then B sets its confirming period for this packet to be $A.R + (A.R - dist(A, B))$ time units. We take the distance of the two hosts into consideration (i.e., $A.R - dist(A, B)$) to avoid bursts of backup retransmission in the case that an expected forward does not indeed occur.

3.2 Advantages of Edge Forwarding

The advantages of Edge Forwarding are as follows:

- *Flooding Reachability*: Unlike the techniques proposed in [9] [10] [11], Edge Forwarding allows a host to drop a flooding packet only when its neighbors can receive the packet from other hosts. In other words, it guarantees any flooding packet to reach all hosts that are reachable through the original plain flooding.

- *Efficiency and Scalability*: Under the proposed strategy, when a host broadcasts, only the recipients close to its partition edges are required to forward the broadcast. This feature eliminates a substantial portion of unnecessary retransmission caused by plain flooding. In comparison with the approach of Flooding with Self-Pruning (FSP) [13], its forwarding rule is much more restricted - a host has to forward a broadcast unless all of its 1-hop neighbors are within 1-hop distance to the broadcast sender. In contrast, Edge Forwarding requires only the hosts near the perimeter of a broadcast radius to rebroadcast, its performance is essentially insensitive to the increases in node density.

- *Control Overhead*: Our technique requires each host to know only the hosts within its 1-hop distance. As a result, our scheme incurs less computation and generates much less control-related network traffic compared to those techniques that require each host to have the knowledge of its 2-hop neighbors (e.g., Dominant Pruning [13], Multipoint Relaying [17], Scalable Broadcast Algorithm [19], etc.). A common approach of tracking 2-hop neighbors is to make each host include all of its 1-hop neighbors in each of its heartbeat broadcasts. While this approach increases the length of heartbeat messages, it also incurs more computational cost. For each heartbeat a host receives, it takes $O(n)$ to update its neighborhood table, where n is the number of hosts that are within 1-hop distance to the broadcast sender. Considering all 2-hop neighbors in making a forwarding decision is also very expensive. Given a uniform host distribution, the number of hosts considered under our scheme is 75% less since a 1-hop coverage is only 25% of a 2-hop coverage. Furthermore, since each host, in our technique, makes the forwarding decision by itself, it avoids the high cost of selecting rebroadcasting hosts. This problem is NP-complete. The heuristic scheme proposed in [17] has a complexity of $m \log n$, where m is the number of selected rebroadcasting hosts and n the number of hosts in the network [17].

- *Implementation*: As mentioned earlier, many routing algorithms have already taken advantage of location information. Typically, these algorithms require each host to know its neighbors within 1-hop distance. Edge Forwarding can be easily incorporated into these techniques to boost their performance.

4 Performance Study

To help understand the performance of Edge Forwarding, we have visualized two flooding techniques. In the visualization, each host is represented by a dot; when a host forwards a data packet, we draw a circle representing its transmission coverage. Figure 5(a) visualizes the performance of Plain Flooding. The circle lines in this figure are very dense because flooding a data packet using this scheme requires every host to forward the data packet. Figure 5(b) visualizes the performance of Advanced Edge Forwarding under the same network snapshot. The much sparse circle lines indicates only a very small portion of hosts are involved in packet retransmission. To investigate the forwarding behavior of this technique, we arrange a host matrix, as showed in Figure 5(c), and let the most left-bottom host initiate a flooding operation. Figure 5(c) shows that the forwarding hosts form some distinct layers, especially in the first three hops. This pattern indicates that forwarding behavior of Edge Forwarding is very close to that of desired *wavefront* forwarding.

In the next, we present the detailed performance study on four reachability-guaranteed flooding techniques: *Plain Flooding* (PF), *Flooding with Self-Pruning* (FSP), *Basic Edge Forwarding* (BEF), and *Advanced Edge Forwarding* (AEF). We focus on their average flooding costs. The cost of flooding a data packet is defined to be the total number of hosts involved in the packet retransmission. The sum of the individual flooding costs is divided by the number of floodings over a simulation run to determine the average flooding cost. We note that FSP requires a host to include its 1-hop neighbors in all packets it broadcasts. This cost is ignored in our performance study. We do not compare with other flooding techniques because they either do not ensure flooding reachability, or require each host to track the changing of network topology beyond 1-hop distance. Nevertheless, it is possible to compare these schemes indirectly with our Edge Forwarding strategy through their performance difference with the plain flooding, which is commonly used as the benchmark for flooding performance.

4.1 Simulation Model

For each simulation run, we generate a certain number of hosts and place them randomly on a square domain. These hosts take turn to flood one data packet using different flooding techniques. Only one flooding occurs at any one time. As we are mainly interested in the reduction in redundant broadcast retransmission, we did not simulate the communication synchronization among the hosts, and assumed that a host could acquire a clear channel whenever it needed. For each flooding, we record the individual flooding costs and compute their average. Since we want to compare the performance of the four different flooding techniques under the same snap shot of the network, we make each host remain static during *one* flooding operation. As the radio transmission propagates at the speed of light, we feel that given the size of the network we simulate, it is reasonable to assume that a flooding packet can be received by all hosts at the same time, although in reality it might pass through several hops. Thus, although we ignore the host mobility during each flooding operation, the flooding costs we collected in each simulation should fairly reflect their true costs in a real network. In particular, the performance difference of these techniques should be quite accurate.

4.2 Simulation Results

In the next subsections, we study how the performance metrics of the proposed techniques are affected by these three parameters: *network area*, *host density*, and *transmission radius*. Table 1 summarizes the parameter values used in the performance study. Roughly, we simulated a small community network, a domain region about 5 to 20 hops with a number of mobile hosts ranging in between 500 to 5000.

4.2.1 Effect of Network Area

In this study, we increased the area of the network region, from 100,000 to 1,000,000 $meter^2$, in each simulation run and fixed the host density at 500 $meter^2/host$. The generated hosts are placed randomly on the network square domain. We fixed the radio transmission radius at 100 meters. The performance data are plotted in Figure 6. We observe that Flooding with Self-Pruning performs almost the same as plain flooding. This confirms that the forwarding rule used in FSP is very restricted - a host can rarely be exempted from forwarding a broadcast as a result of that all of its 1-hop neighbors are already covered by the broadcast transmission. In contrast, the flooding costs under both Edge Forwarding techniques are just a small fraction of that under plain flooding. In particular, their performance gaps become large with the increase of the network domain area. For example, when the network area is 100,000 $meter^2$ (i.e., 500 hosts), the basic version of Edge Forwarding incurs about 50% of the sending cost than the plain flooding; when the network area increases to 1,000,000 $meter^2$ (i.e., 5000 hosts), the flooding cost under the basic Edge Forwarding is only about 25% of that under the plain flooding. This indicates that both Edge Forwarding schemes are highly scalable with respect to the size of network domain. As for the performance comparison between the two Edge Forwarding schemes, the figures show that the advanced version consistently outperforms its basic counterpart. Obviously, this is due to the less-restricted forwarding rule it uses.

4.2.2 Effect of Host Density

In this study, we fixed the radio transmission radius at 100 meters and generated a certain number of hosts, from 500 to 5000, in each simulation run and placed them randomly on a square region of 500,000 $meter^2$. In other words, we reduced the host density from average 100 to 1000 $meters^2/host$. The performance data are plotted in Figure 7. Again, we observe that the performance of FSP is almost the same as that of the plain flooding; and in contrast, both Edge Forwarding schemes can effectively reduce the redundant broadcast retransmission and outperform the plain flooding many times. Notably, we observe that the percentage of the performance improvement becomes large and large with increasing host density. For example, when the host density is 1000 $meter^2/host$, the sending cost under the basic scheme is about 50% of that under the plain flooding; when the host density is increased to 100 $meter^2/host$, the corresponding sending cost is reduced to about 10%. This indicates that the performance of Edge Forwarding solution is not very sensitive to the host density. This characteristic can be explained as follows. First, the number of broadcast retransmission required to cover a network with a fixed domain area is largely not affected by the host density. Second, under Edge Forwarding, only the hosts close to some partition edge are required to forward a broadcast. As a result, the performance of Edge Forwarding solution is not very sensitive to the host density. We note that this performance study, together with the previous one, indicate that our new techniques are particularly suitable for flooding in large scale mobile networks.

4.2.3 Effect of Transmission Radius

In this study, we varied the radio transmission radius, from 50 to 150 meters, in each simulation run and randomly placed 1000 hosts on a square region of 500,000 $meter^2$. The simulation results are plotted in Figure 8. The curve of the send cost for the plain flooding is flat since each host has to forward a flooding packet once and we have the same number of hosts in each simulation run. As for Flooding with Self-Pruning, the figures again show that it cannot reduce the redundant rebroadcast effectively - under all scenarios, its flooding cost is very close to that of plain flooding. On the contrary, both Edge Forwarding schemes incur significantly less flooding cost with a larger transmission radius. This can be explained intuitively as follows. When the transmission range is very small, very few hosts can be found in each of its transmission partitions. In the worst case, all hosts would find themselves close to some partition edge and have to forward a broadcast. As its transmission coverage enlarges, more and more hosts will be contained in each of its partitions. Thus, there is more chance that a host can find itself surrounded by some other hosts within the same partition and does not have to forward a broadcast.

5 Concluding Remarks

We have presented an efficient and low-cost flooding strategy based on the partitioning of each mobile host's radio broadcast coverage. We call the new technique as *Edge Forwarding*, alluding to the fact that it allows mobile hosts to be exempted from forwarding a broadcast unless they are close to some partition edge of the broadcast coverage. In our solution, each host can determine on its own whether or not it should forward a broadcast. Our technique allows a host to drop off a packet only when its 1-hop neighbors can receive the same packet from other hosts. Thus, the new scheme guarantees that a flooding packet is able to reach all hosts that are not isolated from the network. Many existing flooding techniques require mobile hosts to keep track of their neighboring hosts within 2-hop distance. In contrast, our technique requires each host to know only its 1-hop neighbors. Therefore, the new technique is more adaptive to the host mobility and incurs much less overhead in terms of control-related network traffic and computation. Since such 1-hop neighborhood information is a prerequisite to many existing routing algorithms, Edge Forwarding can be incorporated into their implementation easily without any additional control overhead. In our performance study, we compare the new technique with two existing flooding schemes using simulation. Our study shows that under all network settings we simualted, our new technique performs many times better.

References

[1] D. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *Mobile Computing*, T. Imielinski and H. F. Korth, Eds., pp. 153–181. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[2] C. E. Perkins, "Ad Hoc On-Demand Distance Vector (AODV) Routing," INTERNET DRAFT - Mobile Ad hoc NETworking (MONET) Working group of the Internet Engineering Task Force (IETF), November 1997.

[3] Z. J. Haas and M. R. Pearlman, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks," INTERNET DRAFT - Mobile Ad hoc NETworking (MONET) Working group of the Internet Engineering Task Force (IETF), November 1997.

[4] Y. Ko and N. Yaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," in *Proc. of MOBICOM'98*, 1998, pp. 66–75.

[5] C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath, "Flooding for Reliable Multicast in Multi-hop Ad Hoc Networks," in *Proc. of the Int'l Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication*, 1999, pp. 64–71.

[6] J. Jetcheva, Y. Hu, D. Maltz, and D. Johnson, "A Simple Protocol for Multicast and Broadcast in Mobile Ad Hoc Networks," Internet Draft: draft-ietf-manet-simple-mbcast-01.txt, July 2001.

[7] ," IEEE802.11 WaveLAN PC Card - User's Guide, p. A-1.

[8] M. Stemm and R. H. Katz, "Measuring and Reducing Energy Consumption of Network Interfaces in Hand-held Devices," *IEICE Transactions on Communications, Special Issue on Mobile Computing*, vol. E80-B, no. 8, pp. 1125–1131, 1997.

[9] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," in *Proc. of MOBICOM'99*, 1999, pp. 151–162.

[10] Y. Tseng, S. Ni, and E. Y. Shih, "Adaptive Approaches to Relieving Broadcast Storms in a Wireless Multihop Mobile Ad Hoc Networks," in *Proc. of ICDCS'01*, 2001, pp. 481–488.

[11] Y. Sasson, D. Cavin, and A. Schiper, "Probabilistic broadcast for flooding in wireless mobile ad hoc networks," in *Swiss Federal Institute of Technology, Technical Report IC/2002/54*, 2002.

[12] M. T. Sun, W. C. Feng, and T. H. Lai, "Location Aided broadcast in wireless ad hoc networks," in *Proc. of GLOBECOM'01*, 2001.

[13] H. Lim and C. Kim, "Multicast Tree Construction and Flooding in Wireless Ad Hoc Networks," in *Proc. of the ACM Int'l Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM)*, August 2000, pp. 61–68.

[14] B. Williams and T. Camp, "Comparision of Broadcasting Techniques for Mobile Ad Hoc Networks," in *Proc. of MOBIHOC'02*, 2002, pp. 914–205.

[15] W. Lou and J. Wu, "On Reducing Broadcast Redundancy in Ad Hoc Wireless Networks," *IEEE Transactions on Mobile Computing*, vol. 1, no. 2, pp. 111–123, 2002.

[16] W. Peng and X. Lu, "AHBP: An Efficient Broadcast Protocol for Mobile Ad Hoc Networks," *Journal of Science and Technology, Beijing, China*, 2002.

[17] A. Laouiti, A. Qayyum, and L. Viennot, "Multipoint relaying: An efficient technique for flooding in mobile wireless networks," in *35th Annual Hawaii International Conference on System Sciences (HICSS'2001)*. 2001, IEEE Computer Society.

[18] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," in *Proc. of MOBICOM'01*, July 2001, pp. 85–96.

[19] W. Peng and X. Lu, "On the Reduction of Broadcast Redundancy in Mobile Ad Hoc Networks," in *Proc. of MOBIHOC'00*, 2000.

[20] I. Stojmenovic, M. Seddigh, and J. Zunic, "Dominating Sets and Neighbor Elimination Based Broadcasting Algorithms in Wireless Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 1, pp. 14–25, January 2002.

[21] J. Sucec and I. Marsic, "An Efficient Distributed Network-Wide Broadcast Algorithm for Mobile Ad Hoc Networks," in *Rutgers University, CAIP Technical Report 248*, September 2000.

[22] J. Wu and H. Li, "On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks," in *Proc. of the 3rd Int'l Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DiaLM)*, 1999, pp. 7–14.

[23] J. Wu and F. Dai, "Broadcasting in Ad Hoc Networks Based on Self-Pruning," in *Proc. of INFOCOM'03*, March 2003.

[24] J. Wu and F. Dai, "A Generic Distributed Broadcast Scheme in Ad Hoc Wireless Networks," in *Proc. of the 23rd IEEE International Conference on Distributed Computing Systems (ICDCS)*, May 2003, pp. 460–467.

[25] J. Wu, "An Enhanced Approach to Determine A Small Forward Node Set Based on Multipoint Relays," in *Proc. of 2003 IEEE Semiannual Vehicular Technology Conference (VTC2003-fall)*, October 2003, p. to appear.

[26] M. Cagalj, J-P. Hubaux, and C. Enz, "Minimum-Engergy Broadcast in All-Wireless Networks: NP-Completeness and Distribution Issues," in *Proc. of MOBICOM'02*, 2002, pp. 172–182.

[27] O. Egecioglu and T. Gonzalez, "Minimum-energy broadcast in simple graphs with limited node power," in *Proc. of IASTED Int'l Conference on Parallel and Distributed Computing and Systems (PDCS'01)*, 2001, pp. 334–338.

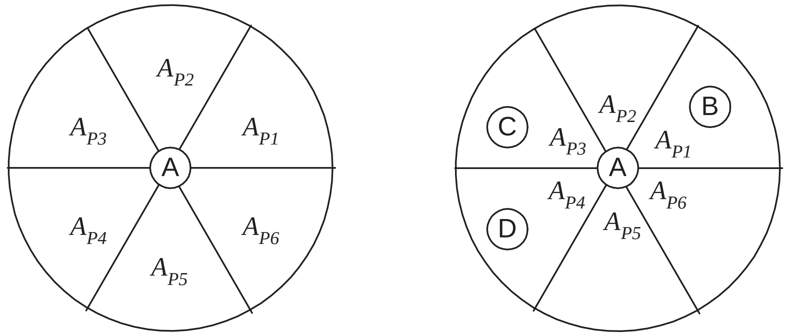
[28] L. Li and I. Nikolaidis, "On minimum-energy broadcasting in all-wireless networks," in *Proc. of the 26th Annual IEEE Conference on Local Computer Networks (LCN'01)*, 2001.

[29] J.E. Wieselthier, G. D. Nguyen, and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," in *Proc. of INFOCOM'00*, 2000, pp. 586–594.

[30] S. Basaggni, I. Chalamtac, V. R. Syrotiuk, and B. A. Woodward, "A Distance Routing Effect Algorithm for Mobility (DREAM)," in *Proc. of MOBICOM'98*, 1998, pp. 76–84.

[31] B. Karp and H. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," in *Proc. of MOBICOM'00*, 2000, pp. 243–254.

[32] J. Li, J. Jannotti, D. Couto, D. R. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing," in *Proc. of MOBICOM'00*, 2000, pp. 120–130.



(a) Partitioning and naming of A's transmission range

(b) B, C, D are A's P_1, P_3, and P_4 neighbors, respectively

Figure 1: Transmission Coverage Partitioning

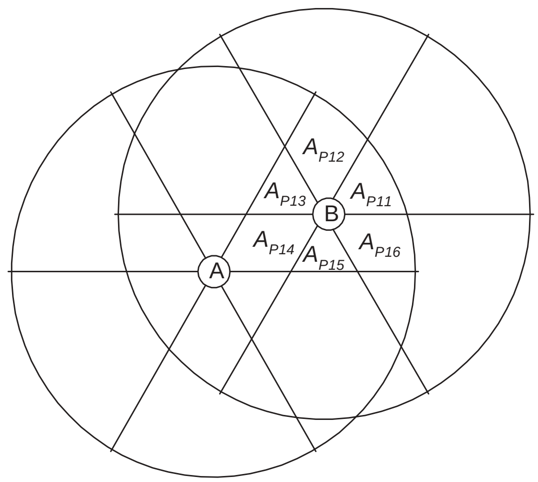


Figure 2: Host B divides A_{P_1} into 6 partitions

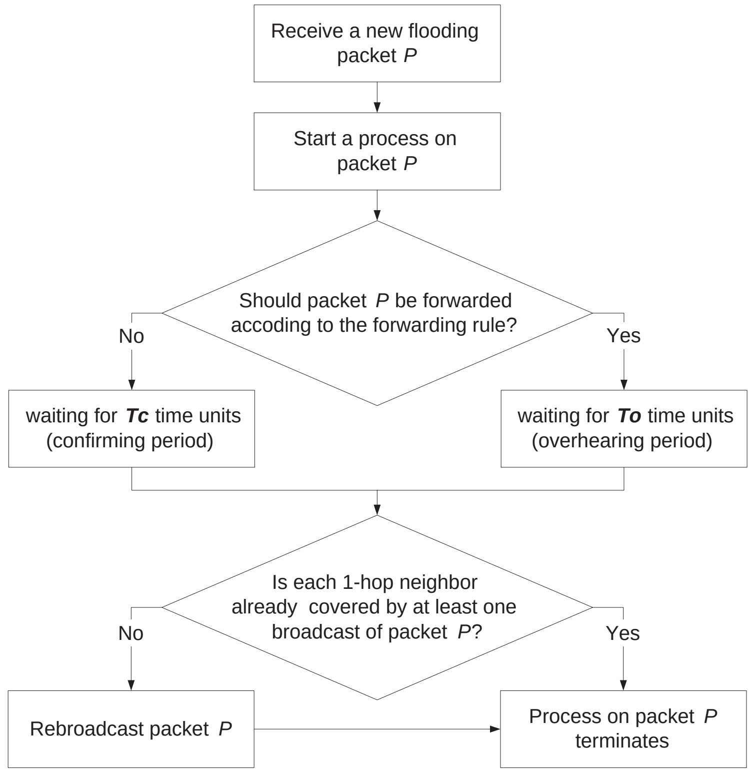


Figure 3: Flowchart of processing a new flooding packet

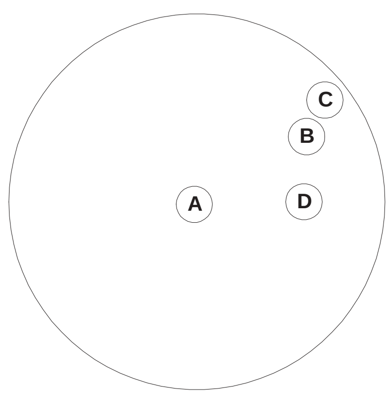
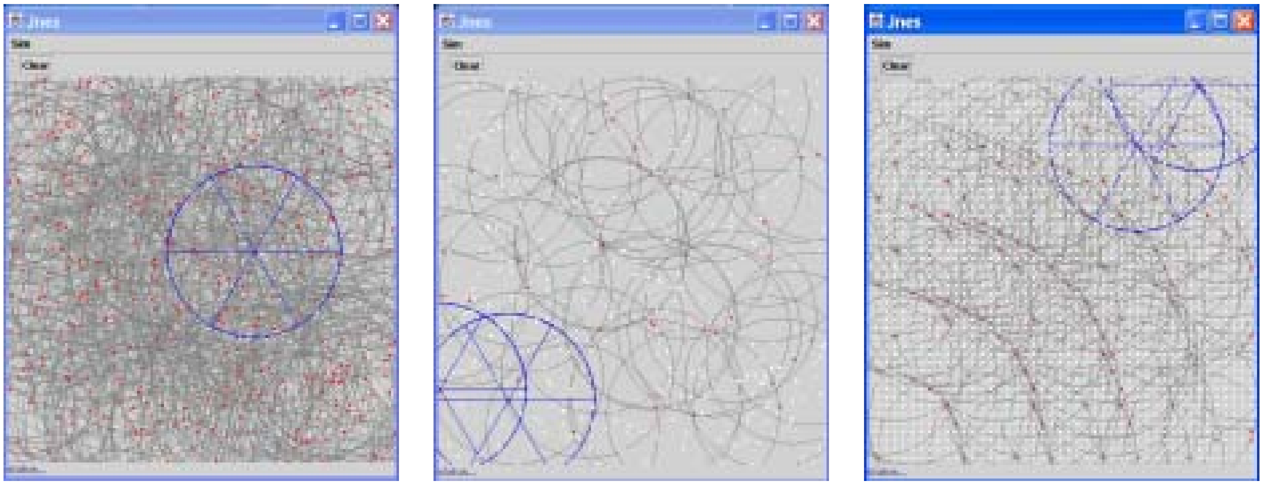


Figure 4: Dynamic setting of overhearing period



(a) Plain Flooding (b) Edge Forwarding (c) Forwarding Pattern

Figure 5: Performance Visualization

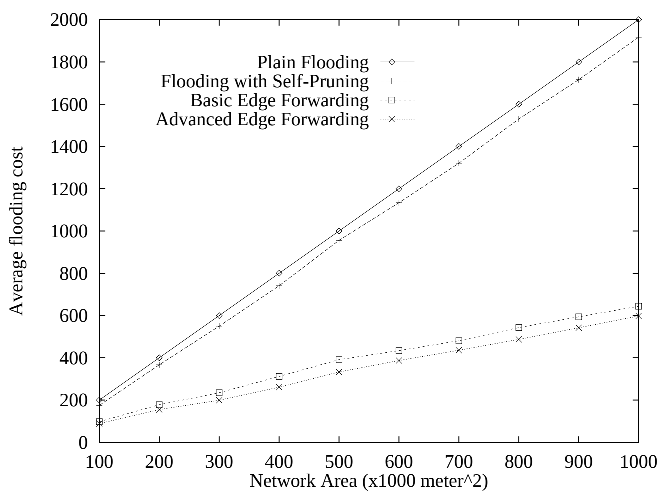


Figure 6: Effect of Network Area

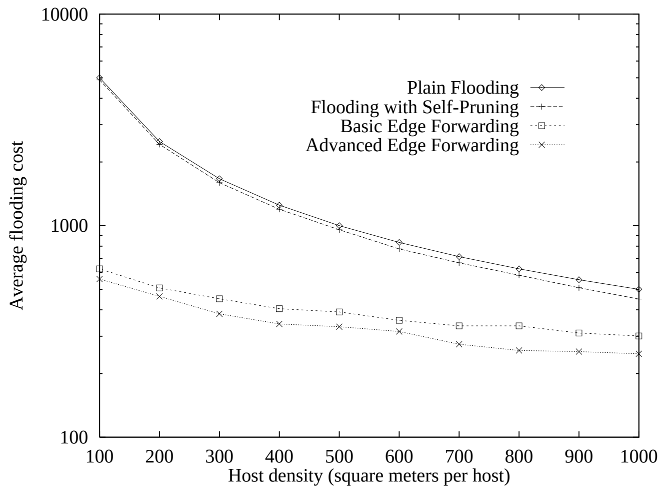


Figure 7: Effect of Host Density

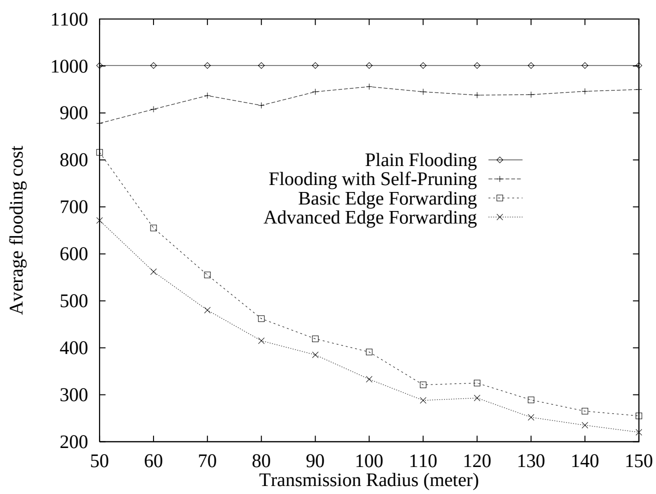


Figure 8: Effect of Transmission Radius

Parameter	default	variation	unit
host density	500	100 - 1000	$meter^2/host$
network area	500,000	100,000 - 1,000,000	$meter^2$
transmission radius	100	50 - 150	$meter$

Table 1: Parameters